

The Monotonic Lagrangian Grid for Rapid Air-Traffic Evaluation

Carolyn Kaplan¹, Johann Dahm^{2*}, Elaine Oran³, Natalia Alexandrov^{4†}, Jay Boris⁵
Naval Research Laboratory, Washington DC 20375

**University of Michigan, Ann Arbor MI 48109*

†NASA Langley Research Center, Hampton VA 23681

The Air Traffic Monotonic Lagrangian Grid (ATMLG) is presented as a tool to evaluate new air traffic system concepts. The model, based on an algorithm called the Monotonic Lagrangian Grid (MLG), can quickly sort, track, and update positions of many aircraft, both on the ground (at airports) and in the air. The underlying data structure is based on the MLG, which is used for sorting and ordering positions and other data needed to describe N moving bodies and their interactions. Aircraft that are close to each other in physical space are always near neighbors in the MLG data arrays, resulting in a fast nearest-neighbor interaction algorithm that scales as N . Recent upgrades to ATMLG include adding blank placeholders within the MLG data structure, which makes it possible to dynamically change the MLG size and also improves the quality of the MLG grid. Additional upgrades include adding FAA flight plan data, such as waypoints and arrival and departure times from the Enhanced Traffic Management System (ETMS), and combining the MLG with the state-of-the-art strategic and tactical conflict detection and resolution algorithms from the NASA-developed Stratway software. In this paper, we present results from our early efforts to couple ATMLG with the Stratway software, and we demonstrate that it can be used to quickly simulate air traffic flow for a very large ETMS dataset.

Nomenclature

N = the number of nodes in the system
 x,y,z = coordinates of the nodes
 i,j,k = indices of the nodes in the data structure
 N_x, N_y, N_z = then number of nodes in directions, x,y,z , respectively

I. Introduction

This paper reports on progress in the continued development of a dynamic air-traffic model, the Air Traffic Monotonic Lagrangian Grid (ATMLG), intended for rapid evaluation of new concepts of operations and methods for control and optimization of transport systems. The model, based on an algorithm called the Monotonic Lagrangian Grid (MLG), can quickly sort, track, and update positions of more than 10000 aircraft, both on the ground and in the air. ATMLG can be used to evaluate new system concepts such as new control strategies for conflict avoidance, separation assurance, and traffic flow management. It may also be used in evaluating the reaction of the system to local and global perturbations, such as weather. More specifically, ATMLG could be used to determine the most efficient way to reroute air traffic after local conditions, such as thunderstorms, have propagated the local disturbance throughout the entire system.

¹ Research Chemical Engineer, Laboratory for Computational Physics & Fluid Dynamics (LCP&FD), Code 6410, AIAA Associate Fellow.

² Student, Department of Aerospace Engineering, AIAA Student Member.

³ Senior Scientist for Reactive Flow Physics, LCP&FD, Code 6404, AIAA Fellow.

⁴ Senior Research Scientist, Aeronautics Systems Analysis Branch, Mail Stop 442, AIAA Associate Fellow.

⁵ Chief Scientist and Director, LCP&FD, Code 6400, AIAA Fellow.

Results from earlier studies showed that the MLG is a promising platform for investigating ways to maintain safe separation between many aircraft in a complex airspace. Previous papers^{1,2} showed the ability of the model to account for and minimize the impact of sudden blockages in flight paths, avoid conflicts, and suggest alternative flight path routes. These simulations kept track of the primary and subsequent conflict avoidance maneuvers necessary to maintain a five mile separation distance between all aircraft, and showed that the number of conflict avoidance maneuvers increases exponentially with the number of aircraft in the volume. These earlier studies,^{1,2} which were intended to be proof-of-concept experiments, used simplified aircraft trajectories and conflict detection and resolution (CD&R) algorithms. We have recently added realistic aircraft trajectories by incorporating flight plan data from the Enhanced Traffic Management System (ETMS). We have also added state-of-the-art CD&R algorithms by linking ATMLG with the NASA-developed Stratway software, which includes a set of conflict resolution algorithms that ensure adequate separation between aircraft. In this paper, we present results from numerical experiments based on these additions.

II. The Monotonic Lagrangian Grid

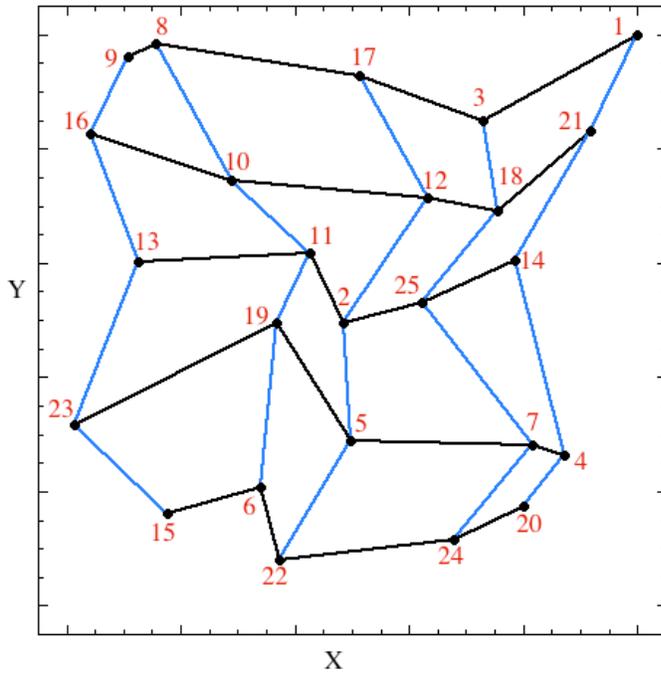
The underlying sorting and ordering algorithm and data structure is the MLG,³⁻⁵ a free-Lagrangian data structure for storing the positions and other data needed to describe N moving bodies. The MLG has been used for two decades as the underpinning for various particle dynamics simulations, including molecular dynamics,⁶ direct simulation Monte Carlo,⁷ and exploratory missile defense⁸⁻¹¹ applications. A three-dimensional MLG data structure in Cartesian coordinates is defined by the constraints:

$$\begin{aligned}
 x(i, j, k) \leq x(i + 1, j, k) & \quad i = 1, \dots, N_x - 1 & \quad j = 1, \dots, N_y & \quad k = 1, \dots, N_z \\
 y(i, j, k) \leq y(i, j + 1, k) & \quad i = 1, \dots, N_x & \quad j = 1, \dots, N_y - 1 & \quad k = 1, \dots, N_z \\
 z(i, j, k) \leq z(i, j, k + 1) & \quad i = 1, \dots, N_x & \quad j = 1, \dots, N_y & \quad k = 1, \dots, N_z - 1
 \end{aligned} \tag{1}$$

where, N_x, N_y and N_z are the number of objects or nodes in each direction., and $N = N_x \cdot N_y \cdot N_z$ is the total number of objects. Here, the meaning of "object" or "node" depends on the particular application. That is, for molecular dynamics simulations, a node may correspond to an atom; for direct simulation Monte Carlo applications, it corresponds to a group of molecules; for the present application, a node or object corresponds to an aircraft. Moreover, a node may be a dimensionless point or it may correspond to a physical object of any appropriate size and characteristics. The constraints mean that each grid line in each spatial direction is forced to be a monotone index mapping. As an example, Figure 1 depicts a small two-dimensional (5x5) MLG. The MLG model developed here is three-dimensional in space; however we are showing a two-dimensional subset to explain the principles of the MLG. The image on the left shows 25 MLG nodes (points in space) at their irregular spatial physical locations, while the table on the right shows the grid indices (in i - j space) of each node. Although the nodes are irregularly spaced (left figure), they are indexed regularly in the MLG by a monotonic mapping between the grid indices and the locations. Note that we generally index the vertical direction from bottom to top (as in a graph), rather than top to bottom (as in a display).

A node with three spatial coordinates has three indices in the MLG data arrays. Data relating to each node are stored in computer memory locations identified by these indices. Thus nodes that are close in physical space are always near neighbors in the MLG data arrays. A computer program based on the MLG data structure does not need to check $N-1$ possible distances to find which nodes are close to a particular node. Rather, the indices of the neighboring nodes are automatically known because the MLG node indices vary monotonically in all directions with the Lagrangian node coordinates. For example, as shown in Figure 1, we automatically know that near neighbors of node 11 are nodes 16, 10, 12, 13, 2, 23, 19, 5, without having to check the distances of all 24 remaining nodes. The cost of most tracking algorithms using the MLG is dominated by the calculation of the interactions of nodes with their near neighbors, and thus the timing scales as N . For applications involving a large number of nodes, such as the air-traffic problem, knowing each node's nearest neighbors automatically, without having to check all $N-1$ distances, results in significant computational savings.

When sorting nodes to MLG order, the simplest MLG algorithm sorts each axis individually. That is, for a 5x5 MLG, it sorts the first five points in the x-direction, then the next five points in the x-direction, etc. After all 25 points have been sorted in the x-direction, it then sorts all 25 points in the y-direction. As each axis becomes sorted, the sorting process may destroy monotonicity in the other axes. Therefore, the sorting process is repeated until all axes are monotonic. The MLG uses two sorting algorithms to put nodes in order. One is a bubble sort, in which



| | | | | | |
|------------|----|----|----|----|----|
| 5 | 9 | 8 | 17 | 3 | 1 |
| 4 | 16 | 10 | 12 | 18 | 21 |
| 3 | 13 | 11 | 2 | 25 | 14 |
| 2 | 23 | 19 | 5 | 7 | 4 |
| 1 | 15 | 6 | 22 | 24 | 20 |
| J/I | 1 | 2 | 3 | 4 | 5 |

Figure 1. An example of a two-dimensional MLG. The figure shows a 2-D MLG (5x5) containing the x- and y-locations of 25 labeled nodes. The solid black (horizontal) lines show the x-links and the dotted blue (vertical) lines show the y-links. The table shows the regular grid indices of the nodes shown in the figure. That is, node-15 is indexed at $i = 1, j = 1$; node-6 is indexed at $i = 2, j = 1$, etc.

each node is compared with the node immediately following it on that axis, and if they are not in monotonic order, their MLG position is switched. This bubble-sorting algorithm is most effective when the nodes are already partially pre-sorted. The other algorithm is a shell sort, in which each node is compared with one that is a half-axis length away, and if the two nodes are not in monotonic order, they are switched. This shell-sorting algorithm is best for completely random data.

The user of the MLG can specify the sort direction (sort from left to right, or from right to left), the axis order (sort x-axis first, then y-axis, or the reverse), and the sorting algorithm (bubble or shell sort). All of these factors affect the specific realization of the MLG that is obtained, because an MLG is not unique.^{12,13} Table 1 shows the number of possible MLGs that can be obtained for a given number of nodes. As shown in the table, even for a small number of nodes, such as 16, there can be up to 405 possible MLGs.

| No. of Nodes | MLG Shape | No. of possible MLGs |
|--------------|-----------|----------------------|
| 4 | 2x2 | 1 - 2 |
| 9 | 3x3 | 3 - 12 |
| 16 | 4x4 | 91 - 405 |
| 25 | 5x5 | 10130 - 97799 |

Table 1. The number of possible MLGs as a function of the number of nodes. The MLG obtained for a given number of nodes is not unique, and some MLGs are of higher quality than others for a given application.^{12,13}

Figure 2 shows three different MLGs based on the same set of node locations. Image (a) shows an MLG obtained when sorting the nodes from an initial random order. This is a valid MLG, as it satisfies the monotonicity constraints in Equation 1. It is, however, an inferior MLG, because it is tangled, and therefore the nodes which are nearest neighbors in i - j space may actually be too far apart in physical space. In order to gauge the quality of an MLG, we have developed a set of diagnostics that are used during a simulation. The diagnostics are based on a dot product which indicates how close an x-link is to the horizontal, and how close a y-link is to the vertical. A

perfectly orthogonal MLG would have a dot product equal to unity for all x-links and y-links. Obviously, any MLG based on real data (such as positions of aircraft in the air) would never form a perfectly orthogonal MLG. However, the MLG algorithm uses a fast grid restructuring technique, called Stochastic Grid Regularization (SGR) to choose an MLG with the highest quality.^{12,13} The SGR is a "simulated annealing" process, in which the nodes are randomly perturbed, and an MLG of the perturbed nodes is used as a starting point for creating an MLG of the original (unperturbed) nodes. The SGR can be applied locally to the tangled parts of the MLG, and it does not significantly increase the cost of the overall calculation. Images (b) and (c) in Figure 2 show how the SGR restructures a tangled grid after one and two SGR iterations.

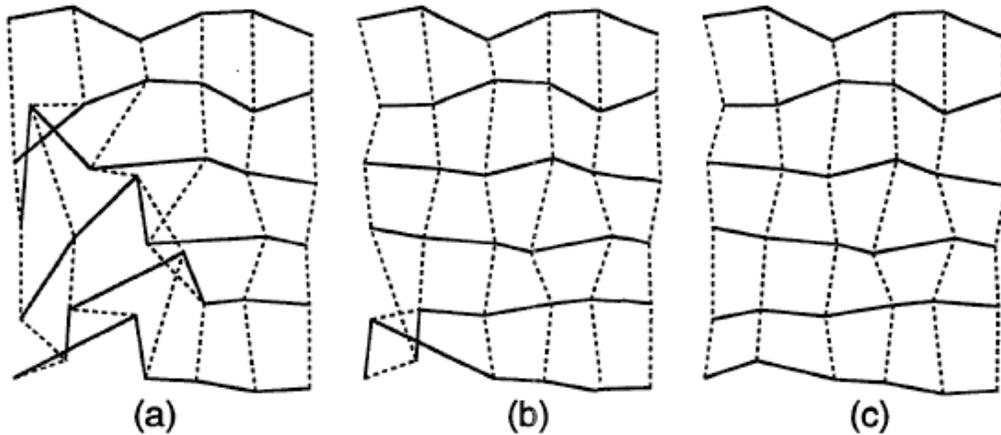


Figure 2. Three different MLGs based on the same set of node locations. Image (a) is an MLG obtained when sorting nodes from a random order. Image (b) is the same MLG after one iteration of SGR, while image (c) is the same MLG after two iterations of SGR. This figure is taken, with permission, from Sinkovits et al.¹²

Each aircraft within the computational domain is represented by one node in the MLG. Since the number of aircraft varies with time (based on flight plan data in the ETMS file), the dimensions of the 3-D MLG must also vary with time, accordingly. Recent upgrades to ATMLG include adding blank placeholders, "holes," within the MLG data structure, which makes it possible to dynamically change the MLG size during the course of the simulation. The addition of holes is shown in Figure 3, which illustrates how a 3x3 MLG can be transformed into a 4x3 MLG. The addition of these blank placeholders is needed when aircraft enter the computational domain. Incorporating holes can also serve as a mechanism to improve the quality of a poorly structured MLG.

An alternative, traditional approach to tracking the movement of aircraft is to partition the space of interest into a grid ("lat-long grid"), with traveling objects moving from one grid cell to another. The location of the background nodes determines the nearness of neighbors. Preliminary computational tests have been conducted on performance comparisons of a simulation with MLG and lat-long grid infrastructure,¹⁴ but definite conclusions are premature. We plan to investigate the relative properties of MLG and lat-long grid in more detail to learn about domains of applicability and a potential for a hybrid approach.

III. Model

At the beginning of the simulation, the computational domain is populated with aircraft from an ETMS flight plan dataset, which includes departure/arrival airport and time, cruising altitude and speed, and a sequence of waypoints between the departure and arrival airports. The dataset contains this information for 169,016 aircraft, over a 72-hour period of time in September 2006. The number of waypoints within each flight varied considerably among the aircraft; i.e., some flights had only a few waypoints, whereas others had more than 10.

During the course of a simulation, varying numbers of aircraft are contained in the computational domain, as some aircraft are in-flight, others have not yet departed, and others have already arrived at their final destination. The latitude and longitude coordinates of the waypoints for each flight are converted to cartesian coordinates through a mercator projection, and the in-flight aircraft are assumed to follow straight-line paths between the waypoints specified in the ETMS file. The location of each aircraft, at any point in time, is calculated based on the speed indicated in the ETMS file, and the amount of time elapsed.

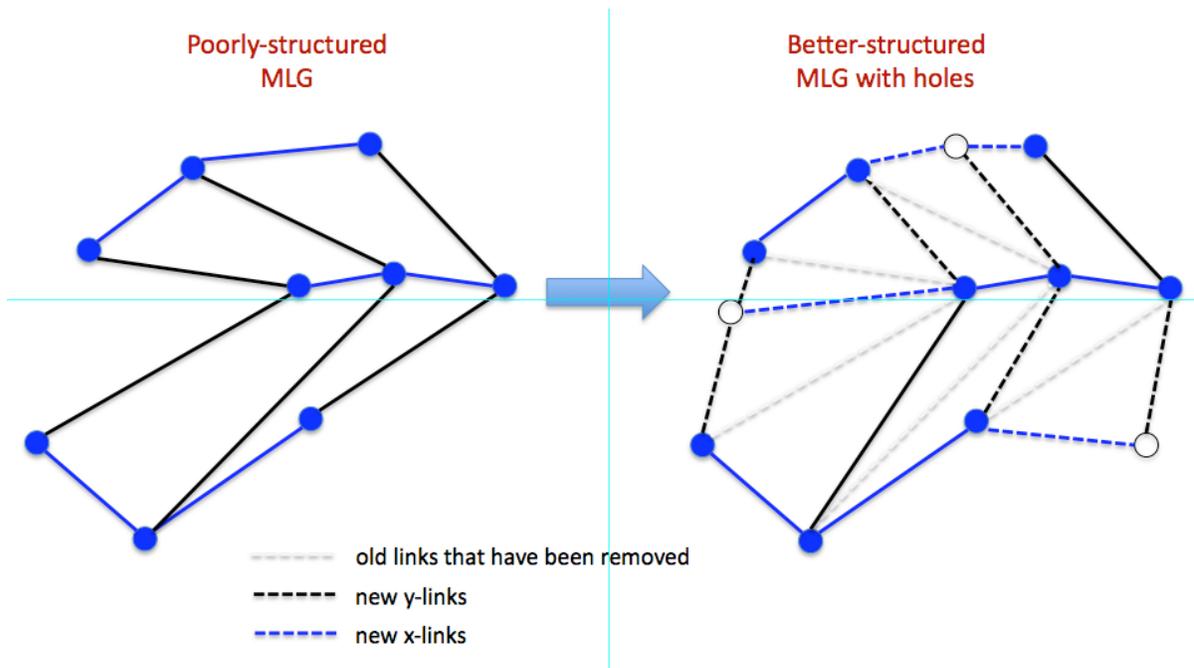


Figure 3. The addition of blank placeholders, or holes, makes it possible to dynamically change the size of the MLG during the course of the simulation. Adding holes is necessary when new aircraft enter the computational domain. Holes also serve to improve the quality of the MLG.

Since the number of aircraft in the computational domain varies with time, the dimensions of the 3-D MLG must also vary accordingly. At each timestep, we compute the number of active aircraft within the domain, and then construct an MLG which is equal to the number of active aircraft plus a sufficient number of blank placeholders. The holes are added so that the total number of nodes in the MLG is an exact multiple of the number of nodes in the x-, y- and z-directions. For example, in a simulation with 3000 aircraft, we can construct a 15 x 15 x 14 MLG, containing 3150 nodes, which would include 3000 real aircraft and 150 blank placeholders.

Two types of CD&R algorithms from the Stratway software package¹⁵ have been implemented in ATMLG. Stratway is a strategic way point adjustment program, which takes a series of flight plans and uses a set of strategies to compute a new flight plan that resolves conflicts.¹⁵ It is an intent-based conflict resolution algorithm, as it runs only on intended flight plan data. It works by taking a single aircraft (the "ownship") and compares its route (based on specified waypoints) with those of the other aircraft (the "traffic") in the volume. If there is a conflict between the ownship and any other traffic aircraft, the planned route of the ownship will change to ensure that there are no conflicts.

The Stratway software package¹⁵ also includes tactical CD&R algorithms. Tactical CD&R is a short-term conflict resolution maneuver that is performed to ensure adequate separation between aircraft. These tactical algorithms are state-based, as they use current position and velocity information to predict whether or not a conflict will occur between a pair of aircraft. Within the Stratway software, we use the CDSS and CRSS algorithms, which calculate pairwise conflict detection and resolution, respectively, given the position and velocity vectors of the ownship and traffic. Multiple conflict resolution solutions are provided based on tracking angle, ground speed and vertical speed.

IV. Results

Over the course of a simulation, aircraft enter, traverse, and leave the computational domain, based on the departure, arrival and waypoint information from the ETMS file. Figure 4 shows a graph of the number of aircraft within the computational domain as a function of time, over the three day timeframe. As shown, during peak traffic, there are approximately 9,000 aircraft in the computational domain. During the calculation, the number of nodes in the MLG increases and decreases similarly.

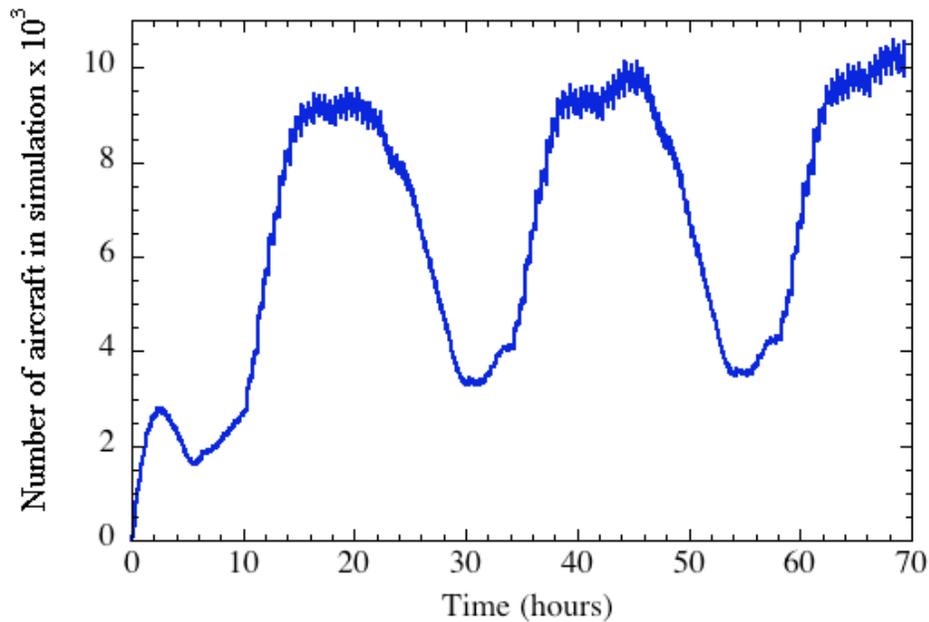


Figure 4. The number of aircraft included in the simulations as a function of time. *The data were obtained from an ETMS file containing 169,016 aircraft, over a three day time period in September 2006. The oscillations are due to the fact the number of aircraft in the computational domain varies with time, as aircraft enter and depart at each timestep. The size of the MLG, which is based on the number of aircraft in the simulation, varies similarly.*

Simulations were conducted using the tactical, pairwise, state-based CD&R algorithm to temporarily change trajectories when aircraft violate the required separation distance. When invoking the tactical resolution algorithm, the user can specify a protection zone distance, which represents the size of the area surrounding an aircraft which cannot be penetrated by any other aircraft. The user can also specify a maximum look-ahead time, which specifies how much forward time the algorithm should consider when predicting conflicts at future locations. In order to test the coupling between the MLG and the Stratway software, a series of calculations were conducted using the first 10,000 aircraft contained in the ETMS file, using various protection zone distances and look-ahead times.

Figure 5 shows the number of conflict avoidance maneuvers required to maintain adequate separation assurance, as a function of the user-specified minimum horizontal separation distance. For example, to maintain a three mile horizontal distance between all 10,000 aircraft in the computational domain, approximately 17,000 maneuvers were required. To maintain separation distances greater than three miles, this figure shows that the number of required maneuvers increases with the size of the protection zone. This is an anticipated result, as one would expect that more maneuvers would be required to maintain larger protection zones. Fig 5 also shows that the number of maneuvers increases again for smaller horizontal separation distances; further investigation is required to understand this trend. As expected, the number of required maneuvers also increases with longer look-ahead time, as shown in Figure 6.

V. Discussion

This work represents our early efforts at coupling ATMLG with the Stratway tactical CD&R software. Further work is needed to better understand the many user-specified input parameters used in Stratway, to ensure that the two codes are properly interfaced. In the results presented here, we varied user-input parameters such as minimum separation distance and look-ahead times; however, we did not vary the filtering parameter, which could significantly change the conflict detection criteria, and hence change the number of conflict maneuvers. Near-term future work will address these issues, and will also include simulations with Stratway's strategic (intent-based) CD&R algorithms.

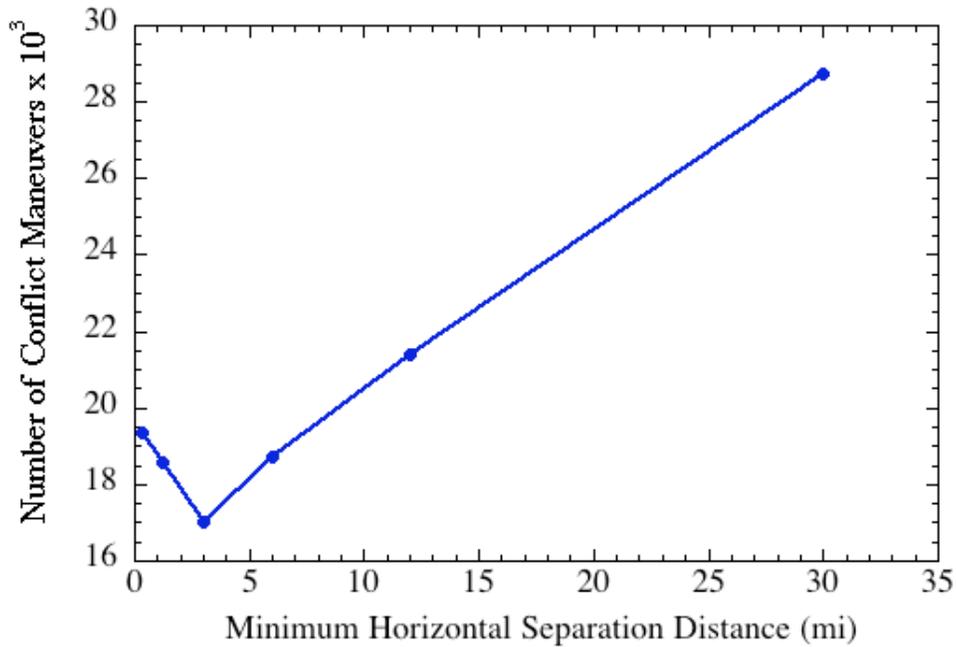


Figure 5. The number of conflict resolution maneuvers required to maintain separation assurance, specified as a minimum horizontal separation distance. Significantly more maneuvers are required to maintain a larger separation between aircraft.

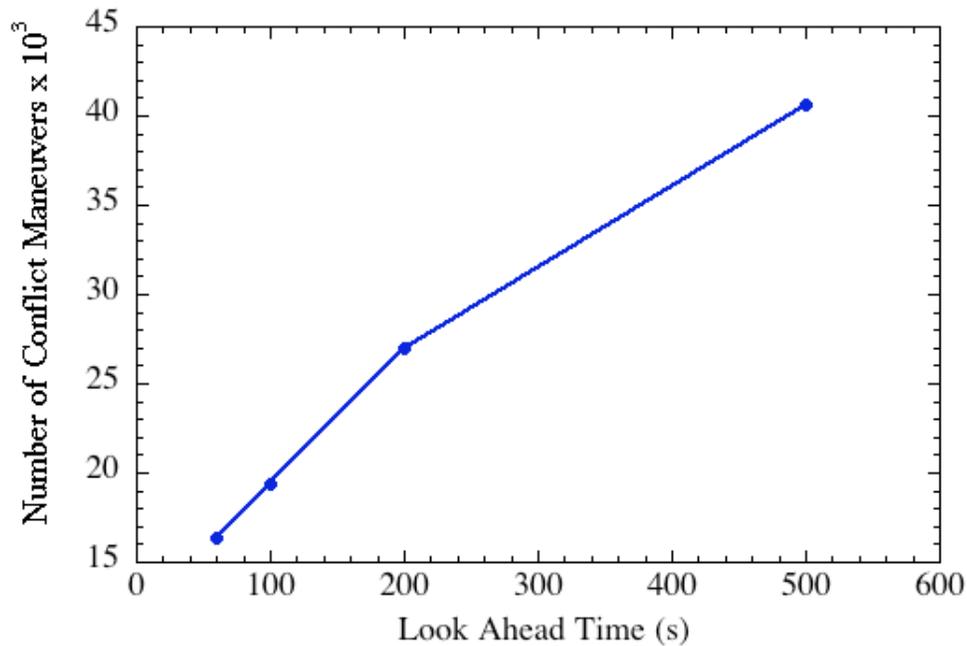


Figure 6. Number of conflict maneuvers as a function of look-ahead time used by the CD&R algorithm

The calculations presented here were conducted using the MLG to locate the two nearest-neighbors in each direction, and then using pairwise tactical CD&R between those near-neighbors only. These simulations were also conducted using the "n-body" approach, in which pairwise tactical CD&R is performed between each aircraft and all

others in the computational domain. For these calculations, the n-body approach requires days on an Intel iMac, while the MLG approach requires only a few hours.

Although use of the MLG approach (compared to the n-body calculation) provides significant computational savings, future work will address the ability of the MLG to catch *all* potential conflicts. As shown in Fig 2, it is possible to obtain an MLG that is tangled, while still satisfying the constraints of Eqn (1). When the MLG is tangled, it is possible to miss nearest-neighbors. Our current strategy to improve the MLG quality, to ensure catching all near neighbors, is by using the simulated annealing process, discussed in Section II. Future work includes optimizing the placement of the blank placeholders, to ensure high quality MLG grids.

Lastly, in this work, we used the MLG to sort based only on the aircraft physical location. The MLG, however, can sort on any quantity that can be expressed by a "greater than" or "less than" rule. Therefore, the MLG could be used to sort on time and physical location, and has the potential for tracking a 4-D (three spatial dimensions and time) history of each aircraft.

Acknowledgments

The authors gratefully acknowledge support from NASA Langley Research Center (LaRC) under Interagency Agreement IA1-910 between LaRC and LCP&FD. The authors also wish to thank Drs. Jeffrey Maddalon, Ricky Butler and George Hagen, from LaRC, for providing the Stratway software, and for their help in implementing it.

References

- ¹Kaplan, C.R., Oran, E.S., Alexandrov, N., Boris, J.P., "The Monotonic Lagrangian Grid Particle Grid: A Fast Tracking Methodology for Air-Traffic Modeling," AIAA Paper 2009-1635, American Institute of Aeronautics and Astronautics, Reston, VA, 2009.
- ²Kaplan, C.R., Oran, E.S., Alexandrov, N., Boris, J.P., "The Monotonic Lagrangian Grid for Fast Air-Traffic Evaluation," AIAA Paper 2010-597, American Institute of Aeronautics and Astronautics, Reston, VA, 2010.
- ³Boris, J., "A Vectorized "Near Neighbors" Algorithm of Order N Using a Monotonic Logical Grid," *J. of Computational Physics*, **66**, 1 (1986).
- ⁴Picone, J.M., Lambrakos, S.G., Boris, J.P. and Jajodia, S., "Initial Comparison of Monotonic Logical Grid and Alternative Data Base Structures," NRL Memorandum Report 5860, Sept 30, 1986.
- ⁵Lambrakos, S.G. and Boris, J.P., "Geometric Properties of the Monotonic Logical Grid Algorithm for Near Neighbor Calculations," *J. of Computational Physics*, **73**, 183 (1987).
- ⁶Lambrakos, S.G., Nagumo, M., Boris, J.P., Oran, E.S., and Gaber, B., "Molecular Dynamics Simulation of a Lipid Bilayer Using Novel Monotonic Logical Grid (MLG) and Multiple Constraint Relaxation (MCR) Algorithms," *Biophysical Journal*, **51**, A440 (1987).
- ⁷Cybyk, B., Oran, E.S., Boris, J.P., and Anderson, J.D., "Combining the Monotonic Lagrangian Grid with a direct simulation Monte Carlo Model," *J. of Computational Physics*, **122**, 323 (1995).
- ⁸Boris, J.P., Picone, J.M., and Lambrakos, S.G., "BEAST: A High-Performance Battle Engagement Area Simulator/Tracker," NRL Memorandum Report 5908, December 31, 1986.
- ⁹Jajodia, S., Meadows, C.A., Boris, J.P., Picone, J.M., and Lambrakos, S.G., "NRL Research in Database Management for the SDI Battle Management System," NRL Memorandum Report 5921, January 30, 1987. Distribution limited.
- ¹⁰Picone, J.M., Boris, J.P., Lambrakos, S.G., Uhlmann, J. and Zuniga, M., "Near-Neighbor Algorithms for Processing Bearing Data," NRL Memorandum Report 6456, May 10, 1989.
- ¹¹Kolbe, R.L., Boris, J.P. and Picone, J.M., "Battle Engagement Area Simulator/Tracker," NRL Memorandum Report 6705, October 8, 1990.
- ¹²Sinkovits, R.S., Oran, E.S. and Boris, J.P., "A Technique for Regularizing the Structure of a Monotonic Lagrangian Grid," *J. of Computational Physics*, **108**, 368 (1993).
- ¹³Sinkovits, R.S., Boris, J.P. and Oran, E.S., "The Stability and Multiplicity of the Monotonic Lagrangian Grid," NRL Memorandum Report 6410-97-7937, 1997.
- ¹⁴Ty Marien, NASA Langley Research Center, oral communication.
- ¹⁵Hagen, G., Butler, R.W. and Maddalon, J.M., Stratway v0.5.1 User Manual, DRAFT 8-5-2010, NASA/TM-2009-000000, August, 2010.